IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Selkirk et al.** § Group Art Unit: **2171**

Serial No.: **09/751,772** § Examiner: **Chen, Te Y.**

Filed: **December 29, 2000** § Attorney Docket No.: **00-062-DSK**

For: **Dynamically Changeable Virtual Mapping Scheme**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 23, 2004.

By: Michele Morrow
Michele Morrow

TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
TRANSMITTED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37);
- Attachment: copy of U.S. Provisional Patent Application No. 60/212,260 filed June 19, 2000; and
- Our return postcard.

A fee of $340.00 is required for filing an Appeal Brief. Please charge this fee to Storage Technology Corporation Deposit Account No. 19-4545. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to Storage Technology Corporation Deposit Account No. 19-4545. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Storage Technology Corporation Deposit Account No. 19-4545.

Respectfully submitted,

Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 385-8777
ATTORNEY FOR APPLICANTS

AF/2171
IFW $

Docket No. 00-062-DSK

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Selkirk et al.** §
§ Group Art Unit: **2171**
Serial No.: **09/751,772** §
§ Examiner: **Chen, Te Y.**
Filed: **December 29, 2000** §
§
For: **Dynamically Changeable Virtual** §
Mapping Scheme §

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

<u>Certificate of Mailing Under 37 C.F.R. § 1.8(a)</u>
I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 23, 2004.

By: Michele Morrow
Michele Morrow

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on September 30, 2004.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party:

Storage Technology Corporation of Louisville, Colorado.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-32.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 1-11, 16, 23, and 29.
2. Claims withdrawn from consideration but not canceled: none.
3. Claims pending: 12-15, 17-22, 24-28, and 30-32.
4. Claims allowed: none.
5. Claims rejected: 12-15, 17-22, 24-28, and 30-32.

C. CLAIMS ON APPEAL

The claims on appeal are: 12-15, 17-22, 24-28, and 30-32.

STATUS OF AMENDMENTS

All of the amendments to the claims have been entered. No after final amendments were made in this case.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 12 - INDEPENDENT

Amended independent claim 12 of the present invention recites a method for generating in a data processing system (Page 11, lines 3-14; and Figure 1) a hierarchical data structure (Figure 8) in a primary storage (Page 27, line 25). The hierarchical data structure includes a plurality of layers arranged according to a hierarchy wherein the plurality of layers includes at least a highest layer and a lowest layer and each layer includes at least one set of data entries. (Page 28, lines 6-18; and Figure 8). Each data entry in each layer represents a range of the virtual address space. (Figures 5, 7, 9 and 10).

Also, each data entry is correlated to a set of data entries in the next lowest layer according to a correlation scheme and each data entry in the lowest layer corresponds to both a virtual address range in the virtual address space and a block address corresponding to a physical data block in the data storage device. (Page 26, line 25 – Page 27, line3; Page 29, line 6 – Page 30, line 21; and Figures 9 and 10).

In addition, each data entry contained within the primary storage corresponds to a virtual address range that is currently occupied with stored data, such that none of the data entries corresponds to only unused physical storage (Page 36, line 4 – Page 38, line 8; and Figure 12) and each physical data block in the data storage device contains virtual address information that identifies at least one corresponding location in the virtual address space for that physical data block (Page 30, line 22 – Page 31, line 8; and Figure 9).

Furthermore, at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer. (Page 5, lines 13-15; Page 25, line 12 – Page 28, line 5; Page 32, line 6 – Page 36, line 3; and Figures 8-11).

B. CLAIMS 19 and 32 - INDEPENDENT

Amended independent method claim 12 of the present invention is representative of amended independent computer program product claim 19 and independent method claim 32. As a result, the claimed subject matter of independent claims 19 and 32 will be found in the same locations as independent claim 12 as laid out above.

C. CLAIM 26 - INDEPENDENT

Amended independent means for claim 26 recites a data processing system for mapping a virtual address space into block addresses of at least one data storage device. The generating means for a hierarchical data structure in the data management system is described on Page 2, line 20 – page 5, line 12 and Figures 1-3. The remaining claimed subject matter of independent claim 26 will be found in the same locations as laid out in Section A above for independent claim 12.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUNDS OF REJECTION (Claims 12-15, 17-22, 24-28, and 30-32).

Claims 12-15, 17-22, 24-28, and 30-32 stand rejected under 35 U.S.C. § 102 (e) as being anticipated by *Selkirk et al* (U.S. Patent No. 6,532,527).

ARGUMENT

A. Ground of Rejection for Claims 12-15, 17-22, 24-28, and 30-32 under 35 U.S.C. § 102 (e).

The Examiner has rejected claims 12-15, 17-22, 24-28, and 30-32 under 35 U.S.C. § 102 (e) as being anticipated by *Selkirk et al* (U.S. Patent No. 6,532,527) (hererinafter *Selkirk* '527). This rejection is respectfully traversed.

The Manual of Patent Examining Procedure (MPEP) § 706.02, V, (B) reads: "[i]f the application is a continuation-in-part of an earlier U.S. application or international application, **any claims in the new application not supported by the specification** and claims **of the parent application have an effective filing date equal to the filing date of the new application**." (Emphasis added). In addition, item (D) of the same section and subsection goes on to read: "[i]f the application properly claims benefit under 35 U.S.C. 119 (e) to a provisional application, the effective filing date is the filing date of the provisional application **for any claims which are fully supported** under the first paragraph of 35 U.S.C. 112 **by the provisional application**." (Emphasis added).

The *Selkirk* '527 application was filed on March 8, 2001, after the filing date of Applicants' present application, which was filed on December 29, 2000. Thus, *Selkirk* '527 is not prior art. But, *Selkirk* '527 does claim benefit to the filing date of corresponding U.S. Provisional Patent Application No. 60/212,260, filed on June, 19, 2000. However, the material Examiner Chen cited in *Selkirk* '527 as prior art against Applicants' current invention is not fully supported by, or found in, Provisional Application 60/212,260. Consequently, the cited material in *Selkirk* '527 is new matter as applied to Provisional Application 60/212,260. Therefore, according to MPEP § 706.02, the new matter will have the *Selkirk* '527 filing date of March 8, 2001 and not the Provisional Application 60/212,260 filing date as proposed by Examiner Chen. As a result, *Selkirk* '527 is not prior art because its cited teachings have an effective filing date which is subsequent to Applicants' current invention's filing date of December 29, 2000.

Provisional Patent Application 60/212,260 teaches a method in an online storage system for dynamically allocating and assigning pointers and associated data structures to provide the indirection needed for the on demand copies. (*Provisional Application* 60/212,260, Page 1, Solution Statement). However, Provisional Application 60/212,260 does not teach or suggest generating in a data processing system a hierarchical data structure in a primary storage wherein the hierarchical data structure includes a plurality of layers arranged according to a hierarchy and wherein at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer as recited in claim 1 of the present invention. Therefore, Provisional Application 60/212,260 is not a proper prior art reference because it does not teach or suggest the recited claim limitations of the Applicants' present invention.

Accordingly, since Provisional Application 60/212,260 does not teach or suggest Applicants' present invention as recited in claim 1, then Applicants respectfully submit that *Selkirk* '527 is not prior art for the features relied upon in the Final Office Action because the earliest possible effective date for these features is March 8, 2001. This is well after the December 29, 2000 filing date of the Applicants' current invention. Consequently, the rejection of claims 12-15, 17-22, 24-28, and 30-32 under 35 U.S.C. § 102 (e) as being anticipated by *Selkirk* '527 is improper and should be withdrawn.

CONCLUSION

In view of the comments above, it is respectfully urged that the rejection of claims 12-15, 17-22, 24-28, and 30-32 not be sustained.

Peter B. Manzo

Peter B. Manzo
Reg. No. 54,700
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The texts of the claims involved in the appeal are:

12. A method for mapping a virtual address space into block addresses of at least one data storage device, the method comprising:

generating in a data processing system a hierarchical data structure in a primary storage;

wherein the hierarchical data structure includes a plurality of layers arranged according to a hierarchy;

wherein the plurality of layers include at least a highest layer and a lowest layer;

wherein each layer in the hierarchical data structure includes at least one set of data entries;

wherein each data entry in each layer represents a range of the virtual address space;

wherein for each layer in the hierarchical data structure for which there exists an next lowest layer, each data entry is correlated to a set of data entries in the next lowest layer according to a correlation scheme;

wherein each data entry in the lowest layer corresponds to both a virtual address range in the virtual address space and a block address corresponding to a physical data block in the at least one data storage device;

wherein each data entry contained within the primary storage corresponds to a virtual address range that is currently occupied with stored data, such that none of the data entries corresponds to only unused physical storage;

wherein each physical data block in the at least one data storage device contains virtual address information that identifies at least one corresponding location in the virtual address space for that physical data block; and

wherein at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer.

13. The method of claim 12, further comprising:

swapping sub-hierarchies of data entries between the hierarchical data structure in primary storage and a secondary storage.

14. The method of claim 12, wherein at least some of the data entries in the lowest layer include a device address identifying an individual storage device in the at least one data storage device.

15. The method of claim 12, wherein the correlation scheme is one of an algorithm, a hash algorithm, a pointer system, and a pointer to correlation logic.

17. The method of claim 12, further comprising:

generating a second data structure, wherein the second data structure identifies exceptional data entries in the hierarchical data structure, wherein each individual exception data entry corresponds to a virtual address range a size that differs from the homogeneous size corresponding to that layer to which the individual exceptional data entry belongs.

18. The method of claim 12, wherein the virtual address information identifies a plurality of corresponding locations in the virtual address space for the physical data block.

19. A computer program product in a computer-readable medium for mapping a virtual address space into block addresses of at least one data storage device, the computer program product comprising:

instructions for generating a hierarchical data structure in a primary storage;

wherein the hierarchical data structure includes a plurality of layers arranged according to a hierarchy;

wherein the plurality of layers include at least a highest layer and a lowest layer;

wherein each layer in the hierarchical data structure includes at least one set of data entries;

wherein each data entry in each layer represents a range of the virtual address space;

wherein for each layer in the hierarchical data structure for which there exists an next lowest layer, each data entry is correlated to a set of data entries in the next lowest layer according to a correlation scheme;

wherein each data entry in the lowest layer corresponds to both a virtual address range in the virtual address space and a block address corresponding to a physical data block in the at least one data storage device;

wherein each data entry contained within the primary storage corresponds to a virtual address range that is currently occupied with stored data, such that none of the data entries corresponds to only unused physical storage;

wherein each physical data block in the at least one data storage device contains virtual address information that identifies at least one corresponding location in the virtual address space for that physical data block; and

wherein at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer.

20. The computer program product of claim 19, further comprising:

instructions for swapping sub-hierarchies of data entries between the hierarchical data structure in primary storage and a secondary storage.

21. The computer program product of claim 19, wherein at least some of the data entries in the lowest layer include a device address identifying an individual storage device in the at least one data storage device.

22. The computer program product of claim 19, wherein the correlation scheme is one of an algorithm, a hash algorithm, a pointer system, and a pointer to correlation logic.

24. The computer program product of claim 19, further comprising:

instructions for generating a second data structure, wherein the second data structure identifies exceptional data entries in the hierarchical data structure, wherein each individual exception data entry corresponds to a virtual address range a size that differs from the homogeneous size corresponding to that layer to which the individual exceptional data entry belongs.

25. The computer program product of claim 19, wherein the virtual address information identifies a plurality of corresponding locations in the virtual address space for the physical data block.

26. A data management system for mapping a virtual address space into block addresses of at least one data storage device, the data management system comprising:

means for generating a hierarchical data structure in a primary storage;

wherein the hierarchical data structure includes a plurality of layers arranged according to a hierarchy;

wherein the plurality of layers include at least a highest layer and a lowest layer;

wherein each layer in the hierarchical data structure includes at least one set of data entries;

wherein each data entry in each layer represents a range of the virtual address space;

wherein for each layer in the hierarchical data structure for which there exists an next lowest layer, each data entry is correlated to a set of data entries in the next lowest layer according to a correlation scheme;

wherein each data entry in the lowest layer corresponds to both a virtual address range in the virtual address space and a block address corresponding to a physical data block in the at least one data storage device;

wherein each data entry contained within the primary storage corresponds to a virtual address range that is currently occupied with stored data, such that none of the data entries corresponds to only unused physical storage;

wherein each physical data block in the at least one data storage device contains virtual address information that identifies at least one corresponding location in the virtual address space for that physical data block; and

wherein at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer.

27. The data management system of claim 26, further comprising:

means for swapping sub-hierarchies of data entries between the hierarchical data structure in primary storage and a secondary storage.

28. The data management system of claim 26, wherein at least some of the data entries in the lowest layer include a device address identifying an individual storage device in the at least one data storage device.

30. The data management system of claim 26, further comprising:

means for generating a second data structure, wherein the second data structure identifies exceptional data entries in the hierarchical data structure, wherein each individual exception data entry corresponds to a virtual address range a size that differs from the homogeneous size corresponding to that layer to which the individual exceptional data entry belongs.

31. The data management system of claim 26, wherein the virtual address information identifies a plurality of corresponding locations in the virtual address space for the physical data block.

32. A method for mapping a virtual address space into block addresses of at least one data storage device, the method comprising:

generating in a data processing system a hierarchical mapping table in a primary storage subsystem;

wherein the hierarchical mapping table includes a plurality of layers arranged according to a hierarchy;

wherein the plurality of layers include at least a highest layer and a lowest layer;

wherein each layer in the hierarchical mapping table includes at least one set of data entries;

wherein each data entry in each layer represents a range within the hierarchical mapping tables;

wherein for each layer in the hierarchical data structure for which there exists an next lowest layer, each data entry is correlated to a set of data entries in the next lowest layer according to a virtual mapping scheme;

wherein each data entry in the lowest layer corresponds to both a mapping table address range in the hierarchical mapping tables and a block entry corresponding to a physical data block in the at least one data storage device;

wherein each data entry contained within a data storage subsystem corresponds to the mapping table address range that is currently occupied with stored data, such that none of the data entries corresponds to only unused physical storage;

wherein each physical data block in the at least one data storage device contains virtual address information that identifies at least one corresponding location in the virtual address space for that physical data block; and

wherein at least some of the data entries in each layer represent virtual address ranges of a homogeneous size corresponding to that layer.

EVIDENCE APPENDIX

A copy of U.S. Provisional Patent Application No. 60/212,260, entitled "System for Providing an Arbitrary Number of on Demand, Unique Copies of Data Utilizing Minimum Storage", filed on June, 19, 2000, is provided for your convenience.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.

CERTIFICATE OF EXPRESS MAILING

I hereby certify that this paper and the documents and/or fees referred to as attached therein are being deposited with the United States Postal Service in an envelope as "Express Mail Post Office to Addressee" service under 37 CFR §1.10, addressed: Box Patent Application, Assistant Commissioner for Patents, Washington, DC 20231.

Express Mail Label No. EL370106355US

Stephanie Klepp
(Type or print name of person mailing paper)

Date of Deposit: June 19, 2000

(Signature of person mailing paper)

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

Address to: **Box PATENT APPLICATION** **Assistant Commissioner for Patents** **Washington, DC 20231**	*Attorney Docket No.*	00-060-DSK
	First Named Inventor: STEPHEN SID SELKIRK	

Sir: This is a request for filing a PROVISIONAL APPLICATION under 37 CFR 1.53(c).

INVENTOR(S)/APPLICANT(S)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND STATE OR FOREIGN COUNTRY)
SELKIRK	STEPHEN	S.	5435 W. 112th Place Broomfield, CO 80020-6802
MILLIGAN	CHARLES	A.	14300 W. 50th Avenue Golden, Colorado 80403

TITLE OF INVENTION

SYSTEM FOR PROVIDING AN ARBITRARY NUMBER OF ON DEMAND, UNIQUE COPIES OF DATA UTILIZING MINIMUM STORAGE

CORRESPONDENCE ADDRESS

Timothy R. Schulte
Wayne P. Bailey
Storage Technology Corporation
One StorageTek Drive, MS-4309
Louisville, Colorado 80028-4309

ENCLOSED APPLICATION PARTS (check all that apply)

__X__ Specification __6__ Number of Pages _____ Small Entity Statement

__X__ Drawing(s) __2__ Number of Sheets _____ Other (specify) _______

__X__ The Commissioner is hereby authorized to charge filling fees or credit any overpayment to Deposit Account No. 19-4545.

Provisional Filing Fee Amount: $150.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

_X_ No

_____ Yes, the name of the U.S. Government agency and the contract number are:

Respectfully submitted,

Date 6/16/00

T Schulte

Name: Timothy R. Schulte
Registration No.: 29,013

_X_ Attorney or agent of record
___ Filed under Rule 34(a)

System for providing an arbitrary number of on demand, unique copies of data utilizing minimum storage

PROBLEM STATEMENT:

Online storage systems provide a few different ways of providing "point in time" or "instant" copies of data. However, these either require extra storage and prior planing (e.g. using mirrored volumes, and breaking the mirror to get the copy) OR they require a complex virtual storage solution with very large mapping tables (e.g. as in the StorageTek Iceberg product line). There is a need for a simpler solution that provides space efficient on demand copies, without the permanent large mapping tables, and without the excess physical storage and pre-planning needed for breaking mirrors.

SOLUTION STATEMENT:

The proposed solution is based on the idea of using dynamically allocated and assigned pointers and associated data structures to provide the indirection needed for the on demand copies. These pointers remain in use only until such time that they are no longer needed due to either (a) the copy is no longer needed, or (b) the two copies are separated and thus can be accessed using algorithmic mapping again.

The pointers and associated data structures are only assigned / allocated when needed to execute an on demand copy operation. In addition, they do not require mapping of a fixed size (e.g., individual volumes) nor do they require the whole storage area being managed to be mapped. Therefore this system will take less mapping table space than a fixed unit map or a log structure map.

Each on demand copy operation causes the initiation of a "mount point" that then describes the copy extents and tracks the status of the copy. There is also a mechanism for tracking exceptions to the mapping within the extent of the mount point (temporary mapping pointers).

The operation of the system when writes and reads occur to a copied extent might include the use of a side file for temporarily storing the data. Several approaches are described, some using side files and some not.

Access to all of the primary data is done via algorithmic mapping, which is very fast. Accesses to areas that are under a mapping for multiple copies are done via a set of pointers that are discovered within the algorithmic function.

Thoughts on "21 Ways to Snapshot" Feb. 3, 2000
S. S. Selkirk

"Ways" brainstorming:

→ broken mirrors < → actual "Fast" Copy
Pre-established
establish & break on ~~[illegible]~~ Sync.

→ copy Old data on write < Pointer Table ± w/o side File

Ⓐ Copy on write to old file Ⓑ Copy on Writ to new ? Ⓒ Copy on both;

→ sidefile: ~~[illegible]~~ • cache old data on write
• Cache New " " "

→ sidefiles: < Temporary in volitile memory
Longterm in non-volitile Memory

→ copy old data on read (no sideFile)
→ Begin copy to new location
→ Writes go to new location
→ Reads, if not yet written to new location, read old location

→ Mirrors → < Volumes
Partitions
Files

→ Mirror Volumes
& do file Snap by break mirror
& Copy file from Mirror &
& re-establish mirrors ...

SO – the copy & copy on read of Old data

- Original File at blocks location A
- Snapshot initiated to location "B".

("B" now considered location of active File!)

- Start Copy operation: Keep Track of Copy progress pointer → ~~B~~ ~~[illegible]~~ ~~[illegible]~~ "C_p"

- On Read ~~to~~ Active File: Read loc. R_1
  - If $R_1 < C_p$, read from $B+R_1$
  - ElseIf $R_1 > C_p$, read from $A+R_1$ OOPS → check exception List
  - ElseIf $R_1 = C_p$, ?
- On Write to Active File: Write loc. W_1
  - If $W_1 < C_p$; Write to $B+W_1$
  - If $W_1 \geq C_p$; Write to $B+W_1$, And Mark $[B+W_1]$ as special case in exception list . . .

  (If $W_1 = C_p \rightarrow ?$)

- On Read ~~to~~ Snap Copy → Read $A+R_2$
- On Write to ~~Snap~~ Snap Copy → W_2
  - If $W_2 < C_p$, Write to $A+W_2$
  - If $W_2 \geq C_p$;
    - Copy $A+W_2 \rightarrow B+W_2$
    - Mark W_2 as exception in list
    - Write ~~to~~ $A+W_2$

3 Feb 2000

Pre- Point in Time Action Required:

	Rd		wrt.	
	Old	new	Old	new
~~B~~Behind copy Point				
beyond copy Point				

	READ		WRITE	
	OLD/SOURCE	NEW/DEST	OLD/SOURCE	NEW/DEST
Op. Loc. < Copy Point (Already copied blks)	<u>Read source</u> ?ck side file Ⓐ if using	<u>Read Dest</u> ? Ck side file Ⓑ if using	<u>Write Source</u> ?ck side file Ⓐ if using	<u>Write Dest</u> ? Ck side file Ⓑ if using
Op. Loc. > Copy Point (Not yet copied blocks)	<u>Read Source</u>	<u>Read Source</u> OR Copy Location, Note in exception list, & Read Dest. OR Iceberg: Copy "Cyl. Blk", then read Destination, & Mark "Cyl. Blk" as copied	Copy Location Source to Dest, Note in exception List, Write Source. <u>OR</u> Write to side File Ⓐ w loc. info. (& copy to Source later).	Write to Dest & add to exception List. <u>OR</u> Write to side file Ⓑ w loc. info. & copy to Dest. later

Op. Loc. < Copy Point

Op. Loc. > Copy Point

June 9, 2000 — check M→MVIT consider: have not copied anything
have copied, but not reconciled

00-060-DSK

6 6/9,

June 9, 2000

Chuck Milligan & Sid Selkirk

39

Yesterday, Thursday, Chuck M. & I talked

→ with algorithmic mapping we still need a de-mark mech. .. to indicate this spot is broken & you have to look elsewhere.

→ Can use the demark mechanism to reduce the 5 step operation to ~~2~~3 step.

5 step Operation:

Write requested to old data

1. Write new blk to side file
2. Rd old data
~~3. Write it to side file~~
3- Write ~~new~~ old block to copy dest.
4. Rd new blk ~~to~~ from side file
5. Write new blk to real location

3 Step
2

1. Rd old
2. Write to copy dest.
3. Write new block

with demark Mechanism

① Demark the old data location & point it to side file
② Write new block to side file

→ works even better if demark can be qualified by virtual Volume ID

Then later there are possible cleanup scenerios:

— new one moved

OR — one demarked is put back

→ Could have multiple writes to demark ptr. locs before cleanup

00-060EDSK

More carts in a drive than 1 cart allows for the Drive to appear to have multiple cart. mounted at one time. (VSM in a box)

Allows ~~for~~ Copy of file from one tape to Another. within single drive

if Drive can execute Disk commands it allows for "ASM in a box"

Can be used as location of snapshot info related to "Disk" activity that is being managed for the customer using ASM technology (ie. Disk front end
- some DASD
- Most of Data on tape)

Can be used to organize localized random activity before Committing data to serial media

10 K cart @ 16 GB = 160 TB of space avail within 5 sec load/mount time with random Acc.
@ 100 GB = 1PB of space total.

at 2X cost of tape system costs

16% is a good cache ratio

$250 media x 10 K = 2.5 m

160 drives @ 20 K .32 m

Lib + others .32 m

$ 3.14 m

Disk costs @ $10/GB = $10 m

Removable & arch in